

REMARKS

Reconsideration of the present application is respectfully requested.

Summary of Rejections

Claims 32-34, 51 and 53-55 stand rejected under 35 U.S.C. § 103(a) based on U.S. Patent no.5,819,292 of Czajkowski et al. ("Czajkowski") in view of U.S. Patent no. 6,453,403 of Hitz et al. ("Hitz"). Claims 25-31, 35, 40 and 49 stand rejected under 35 U.S.C. § 103(a) based on Czajkowski in view of U.S. Patent no. 6,272,502 of Lieuwen et al. ("Lieuwen") and further in view of Hitz. Claims 43-48 and 50 stand rejected under 35 U.S.C. § 103(a) based on Hitz in view of Czajkowski and further in view of Lieuwen.

Interview Summary

A telephonic interview was conducted between Primary Examiner Jean Corrielus and Examiner Miranda Le and Applicant's representative (the undersigned) on 9/18/2007. Claims 32 and 43 were discussed. In particular, Applicant explained why the Examiner's interpretation of Czajkowski as stated in the present Office Action is incorrect and how these claims distinguish over the cited combination (as substantially repeated below). Primary Examiner Corrielus agreed that the amendments presented above would place the present application in condition for allowance.

Applicant maintains that these amendments are not necessitated by the cited art (as explained during the interview and in the remarks below); however, Applicant has nonetheless chosen to amend the claims to expedite allowance of this application.

Status of Amendments

Claims 1-24 and 36-39 were previously canceled.

In this amendment, claims 25, 32, 35, 43 and 51 have been amended, and claims 26, 44, 47, 49 and 50 have been canceled. No claims have been added in this amendment. No new matter has been added.

Therefore, claims 25, 27-35, 40-43, 45, 46, 48 and 51-55 are now pending.

Response to Rejections

Independent claims 25, 32, 35, 43 and 51

Claim 43 is representative of claims 25, 32, 35, 43 and 51 for purposes of discussing the present rejections. Claim 43, as amended, recites:

43. (Currently amended) A method comprising:
maintaining a plurality of persistent point-in-time images of a file system, each persistent point-in-time image representing a state of said file system at a particular point in time, **each persistent point-in-time image having associated therewith a separate map indicating in-use blocks and free blocks of the file system at the corresponding point in time;**
generating a summary map **as a logical OR of said maps associated with at least two of said persistent point-in-time images;** and
making write allocation decisions based on said summary map and a map indicating in-use blocks and free blocks associated with a current state of the file system, including using the summary map to avoid overwriting blocks used by a snapshot. (Emphasis added.)

The cited combination of references neither discloses nor suggest such a method. In particular, the cited combination does not disclose or suggest generating a summary map as a logical OR of maps associated with at least two persistent point-in-time images, where each said map indicates in-use blocks and free blocks of the file system at the corresponding point in time.

The Examiner acknowledges that Hitz does not disclose this feature. However the Examiner cites Czajkowski for such disclosure. Specifically, the Examiner contends that "a logical OR has been applied to the two blocks 317, 318 from the Heap Snapshot 304 to compute the Heap Snapshot 306 such as either block 317 OR block 318 is un-free, both blocks 317, 318 are marked as un-free as shown in Fig. 3, column 7, lines 39-57" (Office Action, page 19).

That contention is incorrect for at least three reasons:

1) Assuming just for the sake of argument that Czajkowski discloses a logical OR in Fig. 3, as the Examiner contends, it could only be a logical OR of two blocks in the same snapshot (e.g. snapshot 304), not a logical OR of maps associated with at least two persistent point-in-time images, as recited in claim 43.

2) Czajkowski does not in fact disclose or suggest any logical OR being performed in relation to Fig. 3. Fig. 3 in Czajkowski shows an example of a memory allocation process (col. 7, lines 17-57), where each of the Heap Snapshots represents a different point in time in the process. The process includes scanning the set of blocks 311-321 from left to right in an effort to locate free (unshaded) blocks that satisfy an allocation request (col. 7, lines 45-46). Anytime an unallocated ("free") block is identified in this process, it is marked as "un-free" (shaded) (col. 7, lines 34-36). This is the only reason why block 317 is marked "free" (unshaded) in Heap Snapshot 304 and then marked "un-free" (shaded) in the next snapshot, 306. There is no logical OR being performed.

Thus, free blocks are marked as "un-free" in this process once they are

identified, regardless of whether or not they satisfy the allocation request. For example, Czajkowski states:

The request fails on a free seventh block 317 because two contiguous blocks were requested, but the fact that the eighth block 318 is already allocated means that the seventh block 317 may only be allocated as a single block. Block 317 is marked unfree and skipped. (Col. 7, lines 49-53)(emphasis added).

Whereas Czajkowski also states, “Finally, the request is filled by the allocation of the ninth block 319 and the 10th block 320, resulting in snapshot 306,” in which blocks 319 and 320 are also both marked as “un-free” (shaded). (Col. 7, lines 53-55)(emphasis added).

Thus, there clearly is no logical OR being performed in this process. All unallocated (free) blocks are marked as “un-free” in the process of Fig. 3 once they are identified.

3) The “Heap Snapshots” mentioned in Fig. 3 of Czajkowski (and related text) do not represent point-in-time images or snapshots that are created by the system, as in the present invention. The “Heap Snapshots” in Fig. 3 are just illustrations of the state of a “heap” at various different points in time, provided to help illustrate a write allocation process step by step. Czajkowski does not disclose or suggest that the system actually creates any actual “snapshots” or point-in-time images of a data set.

In summary, therefore, Czajkowski clearly does not disclose or suggest generating a summary map as a logical OR of maps associated with at least two snapshots, or persistent point-in-time images, where each said map indicates in-use blocks and free blocks of the file system at the corresponding point in time. Likewise,

this feature is also not disclosed or suggested in any of the other cited references.

Therefore, claim 43 in all claims which depend on it are thought to be patentable over the cited art.

Claims 25, 32, 35 and 51 also recite a limitation similar to that discussed above and are therefore thought to be patentable, along with their depending claims, for similar reasons.

Independent claim 32

The above arguments also apply to claim 32.

In addition, in contrast with claim 32, the cited art also does not disclose or suggest that deleting a particular snapshot involves, for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.

The Examiner contends that Czajkowski discloses “indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot in the snapshot just after said particular snapshot (*i.e. Fig. 3 show [sic] the block 321 is unused just after Heap Snapshot 304*).” Office Action, page 4 (emphasis in original).

The Examiner acknowledges that “Czajkowski does not specifically teach:

“receiving a request to delete a particular snapshot; and
“deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot.”
Office Action, page 4.

However, the Examiner contends that Hitz discloses these limitations. That contention is wrong for at least three reasons:

1) The state of block 321 (or any other block) in Czajkowski's Fig. 3 does not depend on any other snapshot, and certainly not on a subsequent snapshot per claim 32. As explained above, the "Heap Snapshots" in Czajkowski's Fig.3 do not represent actual point-in-time images created by the system; they are hypothetical illustrations used only to illustrate the allocation process. Further, as explained above, the free (unallocated) blocks are simply marked as un-free during the write allocation process of Fig. 3, whereas un-free blocks remain marked un-free. There is no dependency between snapshots with regard to the indicated states of the blocks.

2) The Examiner has parsed the above-emphasized claim language in an impermissible manner, effectively negating the limitation's meaning by linguistically splitting a single concept into two parts, and then finding each part in a different reference. That single concept is:

deleting a particular snapshot involves, for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.

The Examiner parses this single claim limitation linguistically into the following two parts:

a) wherein said deleting involves, for a block used by said particular snapshot

b) indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot in the snapshot just after said particular snapshot.

The Examiner alleges that Hitz discloses part a) and Czajkowski discloses part b). See Office Action, p. 4.

Applicants respectfully submit that this parsing of the limitation is nonsensical and essentially renders the limitation meaningless. In fact, part a), taken by itself, does not even make sense or have any clear meaning. Applicants recognize that different limitations in a claim can be found in separate references under an obviousness analysis. However, it is improper to take a single concept, or single limitation, and split it up linguistically in a way that either negates the limitation's meaning or so that any of its parts has no clear meaning by itself.

3) Contrary to the Examiner's contention, there is no disclosure or suggestion in Czajkowski that any act of "indicating said block is free . . ." is associated with deletion of a snapshot, per claim 32.

In summary, therefore, as to claim 32, the cited combination does not disclose or suggest that deleting a particular snapshot involves, for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot. For these additional reasons, therefore, claim 32 and all claims which depend on it are thought to be patentable over the cited art.

Dependent Claims

In view of the above remarks, a specific discussion of the dependent claims is

considered to be unnecessary. Therefore, Applicants' silence regarding any dependent claim is not to be interpreted as agreement with, or acquiescence to, the rejection of such claim or as waiving any argument regarding that claim.

Request for Telephone Interview

If the Examiner does not find the present application to be allowable after considering this response, Applicants respectfully request that the Examiner contact the undersigned at (408) 720-8300 to schedule a telephone interview, before taking further action on this application.

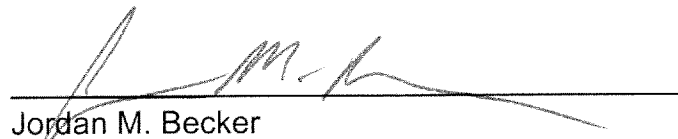
Conclusion

For the foregoing reasons, the present application is believed to be in condition for allowance, and such action is earnestly requested.

If there are any additional charges/credits, please charge/credit our deposit account no. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: October 18, 2007



Jordan M. Becker
Reg. No. 39,602

Customer No. 48102
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
(408) 720-8300